**САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО**

**ИНСТИТУТ КОМПЬЮТЕРНЫХ НАУК И КИБЕРБЕЗОПАСНОСТИ**

**ВЫСШАЯ ШКОЛА ПРОГРАММНОЙ ИНЖЕНЕРИИ**

**Индивидуальное задание**

**«Рекурсия»**

**Вариант 11**

**по предмету «Алгоритмизация и программирование»**

**Выполнил: студент гр. 5130904/30002 Севостьянова А.В.**

**Руководитель Череповский Д.К.**

**Санкт-Петербург**

**2023 г.**

Оглавление

[1. Общая постановка задачи 3](#_Toc150473836)

[2. Детальные требования и тест план 4](#_Toc150473837)

[Таблица с детальными требованиями и тест планом 4](#_Toc150473838)

[3. Программа 5](#_Toc150473839)

[Приложение 7](#_Toc150473840)

[Вывод 8](#_Toc150473841)

# Общая постановка задачи

Написать функцию, выполняющую проверку соответствия строки, заданной параметром, правилу. Функция должна возвращать значение TRUE, если строка соответствует правилу, и FALSE, если строка не соответствует правилу.

***Входные данные функции:*** строка, содержащая проверяемое выражение

***Выходные данные функции:***

* **true**, если выражение соответствует заданному правилу
* **false**, если выражение не соответствует заданному правилу

***Выходные данные программы***: сообщение, несущее информацию о соответствии выражения заданному правилу.

***Понятие выражение определено следующим образом:***

<выражение>::= <терм> | <терм> +<выражение> | <терм> – <выражение>

<терм>::= <множитель> | <множитель> \* <терм>| <множитель> / <терм>

<множитель>::= <целое без знака> | <идентификатор>

<целое без знака>::= <цифра> | <цифра> <целое без знака>

идентификатор>::= <буква>

<цифра> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

<буква>::= a | b | c | d | e

# Детальные требования и тест план

Константное выражение, определенное в теле программы, задано корректно → вызов функции, осуществляющей проверку соответствия правилу.

## *Таблица с детальными требованиями и тест планом*

|  |  |  |  |
| --- | --- | --- | --- |
| Требования | Детальные требования | Данные | Ожидаемый результат |
| 1. ***Выражение (str) задано корректно*** | | | |
| 1. EXPRESSION задано корректно | Константное выражение, определенное в теле программы, задано корректно → вызов функции, осуществляющей проверку соответствия правилу. | 1+2-7\*2/3 | Сообщение: «Выражение соответствует правилу»  Завершение работы программы. |
| 2. EXPRESSION задано корректно | Константное выражение, определенное в теле программы, задано корректно → вызов функции, осуществляющей проверку соответствия правилу. | 1ad+2-7\*2/3 | Сообщение: «Выражение не соответствует правилу»  Завершение работы программы. |

# Программа

***main.cpp***

#include<iostream>

#include"functions.h"

int main()

{

setlocale(LC\_ALL, "ru");

const string str = "1ad+2-7\*2/3"; //проверяемаое выражение

if (isExpression(str)) cout << "Выражение соответствует правилу\n";

else cout << "Выражение не соответствует правилу\n";

return 0;

}

***functions.h***

#pragma once

#include<string>

#include<iostream>

using namespace std;

bool isLetter(const char &symbol);

bool isDigit(const char &symbol);

bool isID(const char &symbol);

bool isUnsigntInt(const string &str);

bool isMultiplier(const string& str);

bool isTerm(const string& str);

bool isExpression(const string& str);

***functions.cpp***

#include "functions.h"

bool isLetter (const char &symbol)

{

const string LETTERS = "abcde";

return (LETTERS.find(symbol) != string::npos);

}

bool isDigit(const char &symbol)

{

const string DIGITS = "0123456789";

return (DIGITS.find(symbol) != string::npos);

}

bool isID(const char &symbol) {return isLetter(symbol);}

bool isUnsignedtInt(const string& str)

{ return (str.length() == 1) ? isDigit(str[0]): (isDigit(str[0]) && isUnsignedtInt(str.substr(1)));}

bool isMultiplier(const string& str)

{ return (str.length() > 1) ? isUnsignedtInt(str) : isUnsignedtInt(str) || isID(str[0]); }

bool isTerm(const string& str)

{

size\_t pos = min(str.find('\*'), str.find('/'));

return (pos == string::npos) ? isMultiplier(str): isMultiplier(str.substr(0, pos)) && isTerm(str.substr(pos+1) );

}

bool isExpression(const string& str)

{

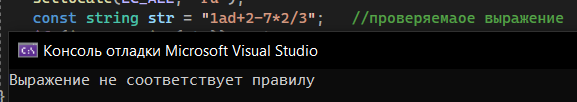
size\_t pos = min(str.find('+'), str.find('-'));

return (pos == string::npos) ? isTerm(str): isTerm(str.substr(0, pos)) && isExpression(str.substr(pos + 1));

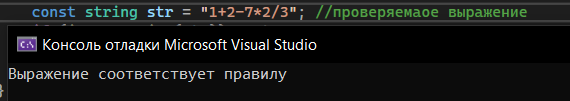
}

# Приложение

*Работа программы*



*Рис. 1 Заданное выражение не соответствует правилу*



*Рис. 2 Заданное выражение соответствует правилу*

# Вывод

В ходе работы над программой:

1. Была изучена рекурсия - определение, описание, изображение какого-либо объекта или процесса внутри самого этого объекта или процесса, то есть ситуация, когда объект является частью самого себя.
2. Выполняемый код был разделен на логические части - функции, что облегчает его восприятие.